NOISE SUPPRESSION OF AUDIO SIGNALS USING WAVELET TOOLBOX IN MATLAB

Arpan Karar
19BEC0584
arpan.karar2019@vitstudent.ac.in
Vellore Institute of Technology
Vellore,India

Abhirup Datta
19BEC0232
abhirup.datta2019@vitstudent.ac.in
Vellore Institute of Technology
Vellore,India

Abstract-Based on the fact that noise and distortion are the main factors that limit the capacity of data transmission in telecommunications and that they also affect the accuracy of the results in the signal measurement systems, whereas, modelling and removing noise and distortions are at the core of theoretical and practical considerations in communications and signal processing. Another important issue here is that, noise reduction and distortion removal are major problems in applications such as; cellular mobile communication, speech recognition, image processing, medical signal processing, radar, sonar, and any other application where the desired signals cannot be isolated from noise and distortion. we will present the development of noise reduction when using wavelet functions in MATLAB. In the foreground, we will demonstrate the usefulness of wavelets to reduce noise in a model system where Gaussian noise is inserted to an audio signal. In the following sections, we will present a practical example of noise reduction in a sinusoidal signal that has been generated in the MATLAB using Wavelet and Communications Toolbox, which it is followed by an example with a real audio signal. Finally, the graph of the noised and denoised signal will be presented.

Keywords—Wavelet, Denoising, white noise, additive white noise

I. INTRODUCTION

Wavelet Toolbox: Provides functions and apps for analysing and synthesizing signals and images. The toolbox includes algorithms for continuous wavelet analysis, wavelet coherence, synchro squeezing, and data-adaptive time-frequency analysis. The toolbox also includes apps and functions for decimated and nondecimated discrete wavelet analysis of signals and images, including wavelet packets and dual-tree transforms.

Using continuous wavelet analysis, you can explore how spectral features evolve over time, identify common time-varying patterns in two signals, and perform time-localized filtering. Using discrete wavelet analysis, you can analyse signals and images at different resolutions to detect changepoints, discontinuities, and other events not readily visible in raw data. You can compare signal statistics on multiple scales, and perform fractal analysis of data to reveal hidden patterns.

Basic Audio Theory: Sound is the vibration of an elastic medium, whether gaseous, liquid or solid. These vibrations are a type of mechanical wave that has the capability to stimulate human ear and to create a sound sensation in the brain. In air, sound is transmitted due to pressure variations at a rate of change that is called frequency. Audio signals, which represent longitudinal variations of pressure in a medium, are converted into electrical signals by piezoelectric transducers. Transducers convert the energy of a mechanical displacement into an electrical signal, either voltage or current. The main

advantage of converting an audio signal into an electrical signal is that the signal can now be processed.

Basic Noise Theory: Noise is defined as an unwanted signal that interferes with the communication or measurement of another signal. A noise itself is an information-bearing signal that conveys information regarding the sources of the noise and the environment in which it propagates.

Signal to noise ratio: SNR describes the total noise present in the output edge detected in an image, in comparison to the noise in the original signal level. SNR is a quality metric and presents a rough calculation of the possibility of false switching; it serves as a mean to compare the relative performance of different implementations. For capturing the wide range of potential *SNR* values and to consider the perception of loudness in humans (in terms of log.), *SNR* is generally given in a logarithmic scale, in decibels (dB) as:

$$SNR = 10.log_{10}(\sigma_x^2/\sigma_e^2)$$

where, σ_x^2 and σ_e^2 are the powers of $x[n]$, and $e[n]$, respectively.

White Noise: White noise is defined as an uncorrelated random noise process with equal power at all frequencies. Random noise has the same power at all frequencies in the range of ∞ it would necessarily need to have infinite power, and it is therefore an only a theoretical concept. However, a band-limited noise process with a flat spectrum covering the frequency range of a band-limited communication system is practically considered a white noise process.

II. METHODOLOGY

With MATLAB, it is possible to process noisy signals containing certain information, such as an audio one, in order to reduce the quantity of noise contained in it. At first, we have taken a sample sine audio wave signal in which we have added a white noise that had been denoised using the following MATLAB code:

```
load wnoisydata; %data is
time table type of format
xnn=table2array(wnoisydata);
xn=xnn(:,1);

xden=wden(xn,'sqtwolog','s',
'mln',5,'sym8');
h1= plot([xn xden]);
h1(2).LineWidth=2;
legend('Original
Signal','Denoised Signal')
```

wden() function : `wden` is a one-dimensional de-noising function. `wden` performs an automatic de-noising process of a one-dimensional signal using wavelets.

`[XD, CXD, LXD] = wden(X,TPTR,SORH,SCAL,N,'wname')` returns a de-noised version `XD` of input signal `X` obtained by thresholding the wavelet coefficients.

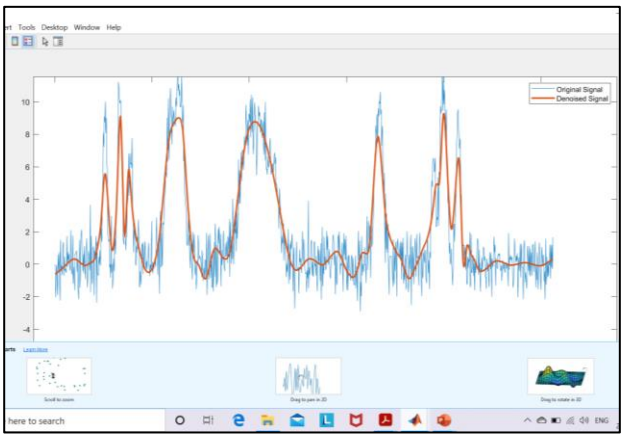


Fig:1. Sample inbuild sinusoidal wave with added noise (using wden function)

With MATLAB, we have used an inbuilt tabular data file, noisdopp and loaded it in our code. Thereafter, we have denoised it

using the wdenoise function. The MATLAB code is as follows:

```
load noisdopp
xden=wdenoise(noisdopp,5);
h1=plot([noisdopp' xden']);
h2(2).LineWidth=2;
legend('Original
Signal','Denoised Signal')
```

wdenoise() function: It performs almost similar function as that of wden() function but the only difference is that it can incorporate Bayes, BlockJS , FDR in its arguments. As a result, it is highly

recommended for denoising audio signals.

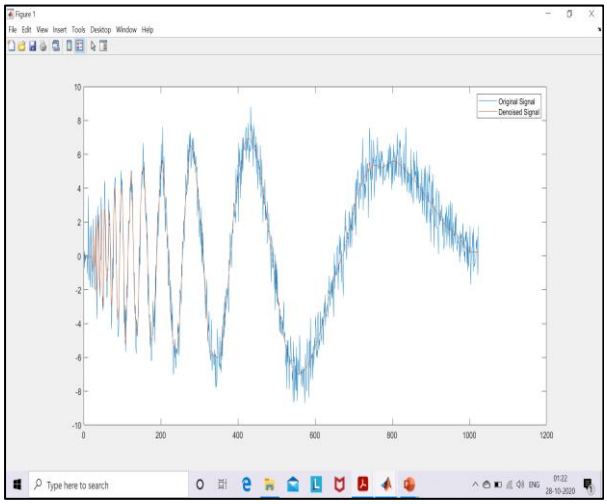


Fig:2. Sample inbuild sinusoidal wave with added noise (using wdenoise function)

MATLAB code for Suppressing a Real Time noise signal: The first thing which we did was collecting a white noise .wav format file from any kind of libraries or external sources. Then, we denoised that noise signal using the wden() function from the Wavelets Toolbox. Thereafter, we plotted the graphs and then used subplots to make the comparison. The MATLAB code we used is as follows:

```
[filename,pathname] =
uigetfile('*.*','Select the
input Audio');
[x,Fs] =
audioread(num2str(filename))
;
xn=awgn(x,15,'measured');
xden =
wden(xn,'sqtwolog','s','mln'
,3,'sym8');
subplot (3,1,1)
plot(x);
title('Original signal');
subplot(3,1,2)
plot(xn,'r');
title('Noisy signal');
subplot(3,1,3)
plot(xden,'g');
title('Denoised signal');
```

In our code, we have used the awgn() function which is used to additive white noise that we had already taken into account by the ‘xn’ variable or the noised variable. Inside the arguments of the wden() function, we have kept the Threshold Estimation Method as ‘sqtwolog’ and applied ‘s’ or Soft Thresholding to our signal. ‘mln’ is used for rescaling of our noise using level dependent estimation methods of level noise with different λ values. Here, the ‘3’ value represents the total no. of decomposition levels of the noise. ‘sym8’ is the name of the wavelet used for the last part of denoising our audio signal. Therefore, we got the wave output as follows:

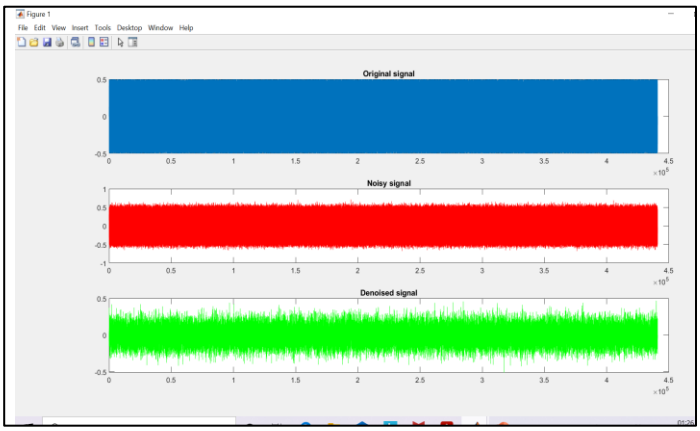


Fig:3. For a whitenoise signal from the gallery with high amplitude of oscillation

Subplot 1: This represents the original whitenoise signal as our input

Subplot 2: This represents the noisy signal which we had taken to superimpose on the information signal.

Subplot 3: This is our final output denoised Signal and can be used further in many purposes.

The Input Audio is:

original_noise.wav

The Output Audio is:

supressed_noise.mp4

III. MATLAB CODE FOR SUPPRESSING A REAL TIME AUDIO SIGNAL (FINAL MATLAB CODE)

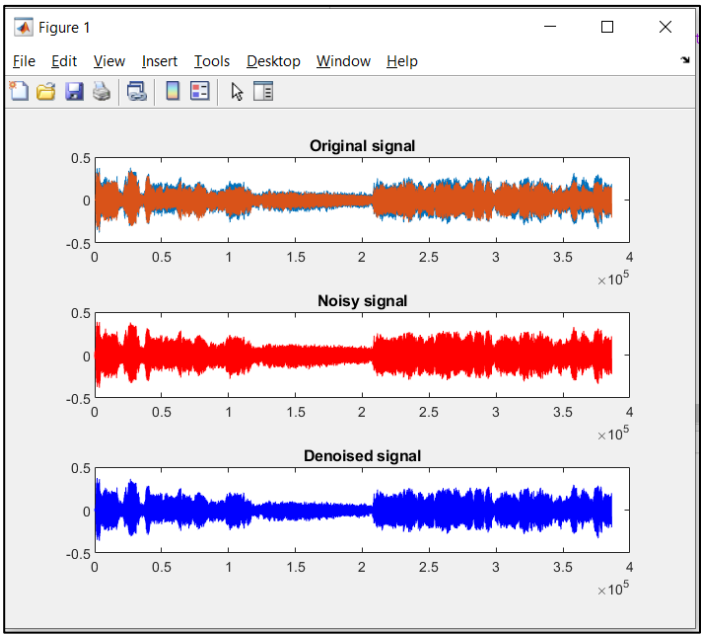
```
[filename,pathname] =
uigetfile('*.*','Select the
input audio');
[x,Fs] =
audioread(num2str(filename))
;
xn = awgn(x,15,'measured');
xden =
wdenoise(xn,'DenoisingMethod
','Bayes','ThresholdRule','S
oft','NoiseEstimate','LevelI
ndependent',8,'Wavelet','sym
8');

subplot(3,1,1)
plot(x);
title('Original signal');
subplot(3,1,2)
plot(xn,'r');
title('Noisy signal');
subplot(3,1,3);
plot(xden,'b');
title('Denoised signal');
```

As mentioned earlier, we have now used wdenoise() function for our audio signal ie a real time song audio signal and then tried to suppress the noise using our MATLAB code to see if it works on an actual signal. We have used a song for this purpose which we had converted from a previously MP4 file to a .wav file for matrix estimation of MATLAB software.

For better clarity in these kinds of signals we have used Bayes method instead of MIMAX or FDR method. As per Threshold rule, we have used a soft thresholding because of only white noise type configuration. Level Independent noise estimation is taken because there are fewer chances of errors if other data don't come into our signal. At last, we have used 8 wavelet decomposition levels for our signal and then again used the same 'sym8' wavelet for final audio processing.

IV. FINAL RESULTS

Thus, our final output waveform is as follows:



We have used the sound() function in our Command Window to hear the noise and the same thing we also did for our previous MATLAB code. For proper

estimation as well as to understand how it is working, we provided sound() function with an argument of 44100 as our ‘Fs’ value.

As per our graphical analysis, the song signal with noise also has been given here:(can be opened using VLC or other media players)

original_music.m4a

Now after denoising our Song with noise suppression the final music is as follows:

denoised_music.m4a

V. CONCLUSION

Thus, after rigorously working and debugging our codes and checking multiple times we came up with the output where our final signal is processed with varying amplitudes and almost a majority of the incoming noise is suppressed by the working model of our project. Thereby we keep forward to make these kind of projects in the future.

VI. REFERENCE

- https://in.mathworks.com/help/images/noise-removal.html
- https://in.mathworks.com/matlabcentral/fileexchange/16646-audio-noise-suppressor
- https://in.mathworks.com/matlabcentral/answers/482699-coding-noise-cancellation-in-m
- https://in.mathworks.com/help/dsp/ug/acoustic-noise-cancellation-lms.html
- https://www.youtube.com/watch?v=JcodkA0eup8
- https://www.youtube.com/watch?v=F_QvT_8kOfc
- https://in.mathworks.com/help/signal/smoothing-and-denoising.html